Threads, SMP, and Microkernels

Chapter 4

Process

• Two characteristics:
  – Resource ownership
    • process includes a virtual address space to hold the process image
  – Scheduling/execution
    • follows an execution path that may be interleaved with other processes
  – These two characteristics are treated independently by the OS
Process

- **process**
  - sometimes referred to as *task* or *job*
  - refers to resource of ownership
  - (addresses the 1st characteristic)
- **thread** or **lightweight process**
  - this is the unit of dispatching
  - (addresses the 2nd characteristic)

Multithreading

- Operating system supports multiple threads of execution within a single process
  - MS-DOS supports a single thread
  - UNIX supports multiple user processes but only supports one thread per process
  - Windows, Solaris, Linux, Mach, OS X, and OS/2 support multiple threads
  - e.g. OS X 10.6 (snow leopard) offers POSIX threads (or pthreads, POSIX 1003.1c standard), and Cocoa threads
Process

- In multithreaded environment a **process** is the unit of resource allocation and a unit of protection
- Processes
  - Have a virtual address space which holds the process image
  - Protected access to processors, other processes, files, and I/O resources

Figure 4.1 Threads and Processes [ANDE97]
Thread

• Within a process there are one or more threads, each with the following:
  – an execution state (running, ready, etc.)
  – a saved thread context when not running
    • may view a thread as an independent program counter operating within a process
  – an execution stack
  – some per-thread static storage for local variables
  – access to the memory & resources of its process
    • all threads of a process share this

Figure 4.2 Single Threaded and Multithreaded Process Models
Benefits of Threads

• Takes less time to create a new thread than a process
• Less time to terminate a thread than a process
• Less time to switch between two threads within the same process
• Since threads within the same process share memory and files, they can communicate with each other without invoking the kernel

Threads in a Single-User Multiprocessing System

• Foreground to background work
  – e.g. spreadsheet, multiple threads display menus, read user input, update spreadsheet etc.
• Asynchronous processing
  – e.g. thead in word processor to periodically flush RAM to disk
Threads in a Single-User Multiprocessing System

• Speed of execution
  – e.g. a process may compute one batch of data while reading in the next.
  – in multiprocessor: true parallel execution of threads in a process

• Modular program structure
  – thread model can be used to “group” activities of process

Quick jump into the real world of research

• Discussion: Processes, Threads and Checkpointing in GRID application

• available at http://www.cs.uidaho.edu/~krings/publications.html